Spring 2017 Cryptography and Network Security

Homework 1

Release Date: 3/15/2017

Due Date: 4/5/2017, 23:59

Instruction

- **Submission Guide:** Please submit all your codes and report to CEIBA. You need to put all of them in a folder named by your student id, compress it to hw1_{student_id}.zip. For example, hw1_r04922456.zip.
- You may encounter new concepts that haven't been taught in class, and thus you're encouraged to discuss with your classmates, search online, ask TAs, etc. However, you must write your own answer and code. Violation of this policy leads to serious consequence.
- You may need to write some programs in the Capture The Flag (CTF) problems. Since you can use any programming language you like, we will use a pseudo extension code**.ext** (e.g., code.py, code.c) when referring to the file name in the problem descriptions.
- You are recommended to provide a brief usage of your code in **readme.txt** (e.g., how to compile, if needed, and execute). You may lose points if TAs can't run your code.
- In each of the Capture The Flag (CTF) problems, you need to find out a flag, which is in `BALSN{...}` format, to prove that you have succeeded in solving the problem.
- Besides the flag, you also need to submit the code you used and a short write-up in the report to get full points.
- In some CTF problems, you need to connect to a given service to get the flag. These services only allow connections from 140.112.0.0/16 and 140.118.0.0/16.

Handwriting

1. CIA (10%)

Please explain three major security requirements: confidentiality, integrity and availability. For each security requirement, please give an example in the real world.

2. Hash Function (10%)

Please explain three properties of a cryptographic hash function: one-wayness, weak collision resistance and strong collision resistance.
For each property, please give an example applied in the real world.

3. Symmetric Cryptography with KDC (15%)

In order to protect the communication privacy of CSIE students, Eric, a new security engineer of the CSIE department, is asked to design a secure communication protocol. The goal of the protocol is simple:
1. Establish a shared secret key S between two legal users (i.e., CSIE students) A and B;
2. A can be assured that B is legal;
3. B can be assured that A is legal.

Note that a student is no longer a legal user after graduation.

To achieve this goal, Eric decides to employ a Key Distribution Center (KDC) in his design. Each legal user i is assigned a shared secret key K_{Si} with the KDC, and can communicate securely with the KDC by the secret key. KDC will only respond to current legal users. Besides, all the keys are securely stored on the server. Here is Eric's design:

$$\begin{aligned} A \rightarrow KDC &: ID_A||ID_B||N_A \\ KDC \rightarrow A &: E_{K_{SA}}(K_S||ID_B||N_A||E_{K_{SB}}(K_S||ID_A)) \\ A \rightarrow B &: E_{K_{SB}}(K_S||ID_A) \\ B \rightarrow A &: E_{K_S}(N_B) \\ A \rightarrow B &: E_{K_S}(f(N_B)) \end{aligned}$$

In the protocol:

$$\begin{aligned} N_A, N_B &: \text{are nonces generated by A and B, respectively.} \\ K_S &: \text{is a session key generated from Alice.} \\ f(N) &: \text{is a simple operation on nonce, e.g., } f(N) = N + 1. \end{aligned}$$

In this problem, you can assume that the KDC is trusted, and the encryption function in the protocol is ideal (i.e., without the key, no one can recover the encrypted message or modify the encrypted message without being detected). We consider a network attacker that can eavesdrop, intercept, and manipulate the communication. The attacker can also be a former student who had a secret key with the KDC but is no longer a legal user.

(a) (5%) What's the purpose of N_A and N_B in this protocol? (Imagine if the protocol doesn't use nonce, what kind of the attack would become possible?)

(b) (5%) How can an attacker break the goal of the protocol?

(c) (5%) Try to fix the protocol to prevent the attacker you described above. Please explain clearly.

Capture The Flag

4. Classical Cipher (6%)

`Classical cipher` is a type of cipher that was used historically but now has fallen. However, it's still informative to learn the design and attacking techniques of these ciphers. In this homework, you are going to play with some well known classical ciphers. You can access the service by `nc 140.112.31.109 10000`. Try to get the flag behind the challenge and explain how you solve the problems in your write-up. Save you script as `code4.ext`.

5. Google can beat this (10%)

SHA1 is an implementation of the cryptographic hash function. In this problem, you first need to find an input x such that the rightmost 24 bits in $SHA1(x)$ are the same as what the service says. And in the second part of the problem, you need to find an input y such that $SHA1(x) == SHA1(y)$. You can access the service by `nc 140.112.31.109 10001` for more information. Please explain how you find x and y and save your code as `code5.ext`. Note that you should send your x and y in `hex-format`.

6. Many-time pad (10%)

> "One-time pad is so secure that I use the same key to encrypt all my message."

Every line of the file `MTP/cipher` contains a valid English sentence encrypted by xoring the same key. One of them is the flag. Please find it out.

Hint: Key length is 64 bytes

7. Backdoor of Diffie Hellman (11%)

> "Hi, I am Alice. I used the script `DH_backdoor/DH.py` to send a top secret flag to my best friend, Bob. However, after sending the flag, I found that the generator g was modified by someone. It must be a backdoor of the Diffie Hellman algorithm. I have collected the parameters we used in the file `DH_backdoor/parameters`. Can you find out whether our flag is leaked?"

Decrypt the variable **cipher** in the file `DH_backdoor/parameters` to get the flag.

Hint: $g_{old}^{\frac{p-1}{691829}} \mod p = g_{backdoor}$ *is this a coincidence?*

8. Man In The Middle (13%)

`Man-in-the-middle attack` is a common attack technique where an attacker can secretly relay and possibly alter the messages between victims while making them beleive they are talking to each other. In the class, we talk about this attack when introducing authentication mechanisms. In this homework, you are asked to perform such an attack to a communication service `MITM/mitm.py`. In the service, the server uses a `Diffie-Hellman protocol` to exchange key information with another end-point. The key is later used to encrypt the flag. Can you recover the flag? You can access the service by `nc 140.112.31.109 10002`. Explain how you perform the attack in your write-up and save your code as `code8.ext.`

Hint1: The value space of each password is small.
Hint2: Have you heard of the reflection attack?

9. Only admin can print flag (15%)

We discussed entity authentication in the class. The verifier can check the claimed identity of the prover, and the prover usually provides a password as a proof of the identity. Given the server `Only/admin.py`, you have one account **guest** and the password **IT'SMEPASSWORD**. But you don't have the administrator password. Can you login as **admin**? You can access the service by `nc 140.112.31.109 10003`. What is the flag? Explain the vulnerability and how you attack. Save your code as `code9.ext`.